

IN THE CLAIMS:

1. (currently amended) An object oriented computing system on a computer platform, comprising:

objects comprising at least one of software components and building blocks with semanticless, dynamically ~~likable~~ linkable inputs and outputs stored on a memory of the computer system; and

an event communication framework providing automated, pattern-based, fully distributable events so that the objects are combined substantially without at least one of changing code and writing adapters.

2. (previously presented) The object oriented computing system of claim 1, wherein the inputs and outputs of the objects are provided via CsaConnectable and CsaRemote objects, respectively.

3. (previously presented) The object oriented computing system of claim 2, wherein each data structure associated with the inputs and outputs is described in a separate header file which can be used by every object to be linked.

4. (previously presented) The object oriented computing system of claim 2, wherein each object is a shared library which is dynamically linkable at runtime by an ASCII configuration filing names of the inputs and outputs of the objects.

5. (currently amended) An object oriented computing system on a computing system, comprising:

a memory of the computing system storing objects;

said objects comprising at least one of software components and building blocks having dynamically ~~likable~~ linkable inputs and outputs and internal tasks for queuing of data transferred into and out from the objects via said inputs and outputs, respectively; and

an event communication framework providing automated, pattern-based, fully distributable events so that the objects are combined substantially without at least one of changing code and writing adapters.

6. (previously presented) The object oriented computing system of claim 5, wherein the inputs and outputs of the objects are provided via CsaConnectable and CsaRemote objects, respectively.

7. (previously presented) The object oriented computing system of claim 6, wherein each data structure associated with the inputs and outputs is described in a separate header file which can be used by every object to be linked.

8. (previously presented) The object oriented computing system of claim 6, wherein each object is a shared library which is dynamically linkable at runtime by an ASCII configuration file containing names of the inputs and outputs of the objects.

9. (currently amended) A method for designing software components in an object oriented computing system, comprising the steps of:

defining input and output events that are fully distributable;

configuring dynamic linkable, semantic-free software modules by input and output connections points and storing the modules on a memory of the computer system; and

providing autorouted pattern based fully distributable events based on an event communication framework so that the modules are combined substantially without at least one of changing code and writing adapters.

10. (currently amended) A storage medium including object oriented code having an object oriented computing system on a computer platform, comprising:

objects comprising at least one of software components and building blocks with semanticless, dynamically linkable inputs and outputs stored in memory of the computer system; and

an event communication framework providing automated, pattern-based, fully distributable events so that the objects are combined substantially without at least one of changing code and writing adapters.

11. (previously presented) The storage medium of claim 10, wherein the inputs and outputs of the objects are provided via CsaConnectable and CsaRemote objects, respectively.

12. (previously presented) The storage medium of claim 11, wherein each data structure associated with the inputs and outputs is described in a separate header file which can be used by every object to be linked.

13. (currently amended) The storage medium of claim 12, wherein each object is a shared library which is dynamically ~~likable~~ linkable at runtime by an ASCII configuration filing names of the inputs and outputs of the objects.

14. (currently amended) A storage medium, comprising:
object oriented code for an object oriented computing system on a computing system;

objects comprising at least one of software components and building blocks stored on a memory of the computer system and having dynamically linkable inputs and outputs and internal tasks for queuing of data transferred into and out from the objects via said inputs and outputs, respectively; and

an event communication framework providing automated, pattern-based, fully distributable events so that the objects are combined substantially without at least one of changing code and writing adapters.

15. (previously presented) The storage medium of claim 14, wherein the inputs and outputs of the objects are provided via CasConnectable and CsaRemote objects, respectively.

16. (previously presented) The storage medium of claim 15, wherein each data structure associated with the inputs and outputs is described in a separate header file which can be used by every object to be linked.

17. (previously presented) The storage medium of claim 15, wherein each object is a shared library which is dynamically linkable at runtime by an ASCII configuration file containing names of the inputs and outputs of the objects.

18. (currently amended) A method for designing software components in an object oriented computing system having a storage medium including object oriented code, comprising the steps of:

defining input and output events that are fully distributable;

configuring dynamic linkable, semantic-free software components by input and output connections points and stored on a memory of the computer system; and

providing autorouted pattern based fully distributable events based on an event communication framework so that the components are combined substantially without at least one of changing code and writing adapters.